International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 22677-22684

http://www.ripublication.com

Design of Pipelined Based Dct Architecture For Image Processing Applications

[1]Abhinaya Rajasekaran, [2]R.Manikandan
[1,2]School of Computing, SASTRA University,Thanjavur-613401, India
[1]Email: abhinaya.rajasekaran@gmail.com

Abstract

Discrete Cosine Transform is considered to be an essential tool in image and video compression systems. Fast algorithms for achieving appropriate approximation of DCT transform have been developed for video processing systems to overcome high energy consumption. Usage of only additions and subtractions in the proposed design has led to substantial decrease in complexity along with area and power consumptions. In this paper a DCT architecture using only 12 additions without the usage of multipliers has been proposed. The proposed architecture is compared with the existing methods for power, area, speed and complexity. The proposed design can be used for video processing with very little reconfiguration.

Keywords: DCT approximation, low complexity, multiplier less, power consumption, image compression, VLSI architecture.

Introduction

Discrete Cosine Transform (DCT) [1] [2] [9] has been an active research area over the last decade and has been used in various image processing applications for reduced bandwidth image and video transmission including JPEG and MPEG standards. DCT algorithm is highly preferred because of its simplicity and symmetrical properties. The necessity of using DCT transform is to remove the inter-pixel redundancy from the original image representation. Reduction of inter-pixel redundancy transforms the image data to a new representation where the average values of transformed data are smaller when compared to the original form. In this way the compression is achieved.

To achieve higher compression ratio the correlation among the image pixels should be maintained high. The following properties should be present in an image transform.

a) Should perform inverse transformation.
b) Should be capable of de-correlating the original image data.
c) Should be able to clearly separate the frequency.

In an image the low frequency contents contain very high visual information when compared to its high frequency counterpart. The high frequency contents denotes very fine details of an image which is not useful in most of the applications as they are not visible to human eyes. Therefore to achieve compression these higher frequency contents can be excluded in the coding stage. The coefficient values are indirectly proportional to the compression ratio. Due to the requirement of fast image compression and complex transforms, computation times are increased making the process slower. So, for an ideal DCT transform maximum energy compaction and low computational complexity are much needed traits.

Existing Method

In this section two existing methods have been discussed. Section 2.1 comprises of Standard DCT method and DCT approximation using 14 additions has been discussed in section 2.2

Standard DCT Method

The Discrete Cosine Transform is a Fourier related transform that converts spatial domain to frequency domain and quantize high frequencies more coarsely. Here a fast DCT based on algorithm-architecture transformations and DIF (decimation-in-frequency) approach is discussed. The DCT is depicted in the following equation [1],

$$X(k) = e(k)\sum_{n=0}^{7} x(n)\cos\left[\frac{(2n+1)\pi k}{16}\right] \quad \text{k=0, 1....7} \tag{1}$$

Where,

$$e(k) = \begin{cases} \frac{1}{\sqrt{2}}, if\ k = 0 \\ 1, otherwise \end{cases} \tag{2}$$

The equation in (1) can be represented in matrix form as follows:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} e_4 & e_4 & e_4 & e_4 & e_4 & e_4 & e_4 & e_4 \\ e_1 & e_3 & e_5 & e_7 & -e_7 & -e_5 & -e_3 & -e_1 \\ e_2 & e_6 & -e_6 & -e_2 & -e_2 & -e_6 & e_6 & e_2 \\ e_3 & -e_7 & -e_1 & -e_5 & e_5 & e_1 & e_7 & -e_3 \\ e_4 & -e_4 & -e_4 & e_4 & e_4 & -e_4 & -e_4 & e_4 \\ e_5 & -e_1 & e_7 & e_3 & -e_3 & -e_7 & -e_1 & -e_5 \\ e_6 & -e_2 & e_2 & -e_6 & -e_6 & e_2 & -e_2 & e_6 \\ e_7 & -e_5 & e_3 & -e_1 & e_1 & -e_3 & e_5 & -e_7 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix}$$

Where,

$$e_i = cos\frac{i\pi}{16} \tag{3}$$

The DCT architecture constructed from the standard matrix is represented below:

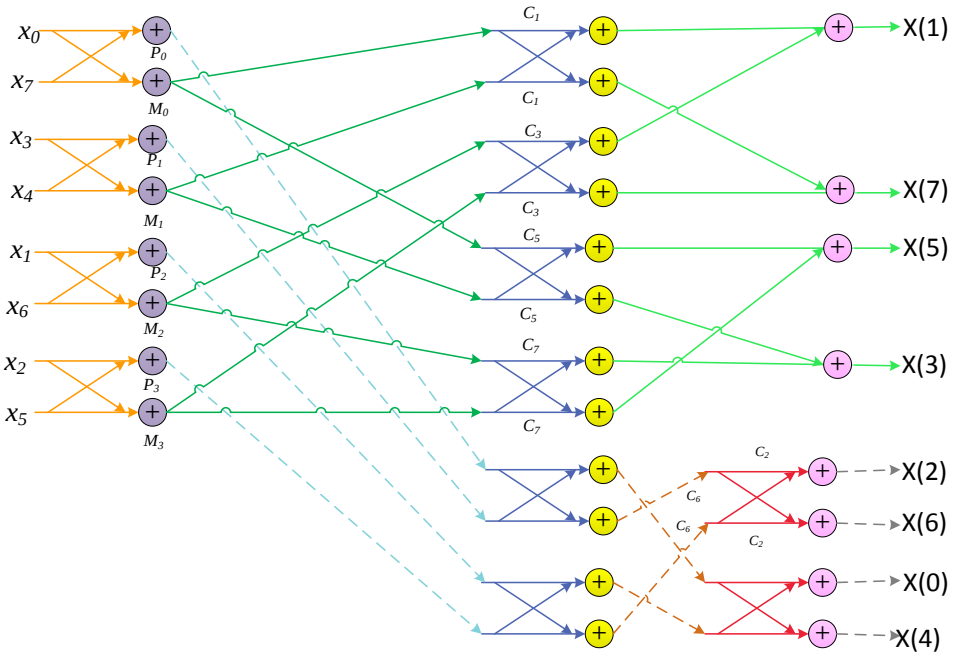

Figure 1: Signal flow diagram of standard DCT [1]

The standard DCT architecture shown in Fig.1 uses a large number of multipliers and adders there by increasing the complexity of the structure. This entails huge power consumption and increase in area as well. The existing method is not only slow but also an energy inefficient system. These are not the traits needed in a VLSI system. Hence a less complex architecture is preferred for high speed and better power management.

DCT Approximation Using 14 Additions

In this method, the standard DCT architecture is modified by scaling the standard DCT matrix C in (3) by 2 and each element is subjected to a round off operation. The matrix C_0 represents the scaled matrix [4] [5].

$$C_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The above matrix C_0 has following interesting properties:

(i) Nearby elements of the matrix is -1, 0 or 1 which indicates that no multiplication is required for the computation.

(ii) The transform of the matrix can be computed using additions without any requirements of bit shift operators.

(iii) It is a Quasi symmetrical tool because an approximate inversion can be performed using the scaled transpose matrix.

The above matrix C_0 can be shown in th e following signal graph

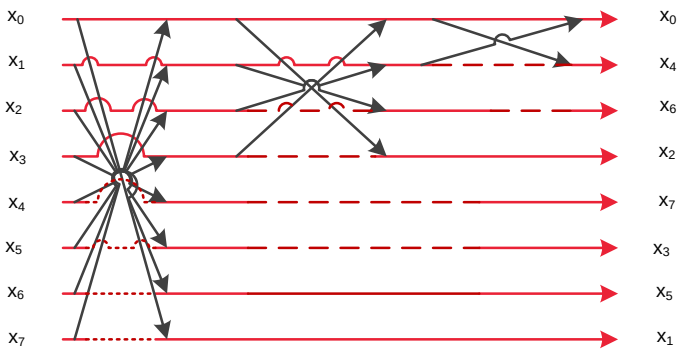

Figure 2: Signal flow diagram of DCT using 14 Additions [4]

Performing $C^T = (1/2)C_0$ achieves a rough approximate of the DCT matrix where (1/2) represents only bit shifts which allows it to outperform SDCT in a wide variety of ways. However this process results in the lack of orthogonality of C_0 since $C_0^{-1} \neq C_0^T$ and also provides poor approximation when compared to other methods. To overcome the lack of orthogonality, the matrix C_0 is multiplied by a diagonal matrix S where $S = \sqrt{(C_0.C_0^T)^{-1}}$ on computing which we get $S = diag\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}},\right)$ This new matrix depicted as C_{or} is orthogonal in nature and inherits useful properties such as low complexity and a diagonal orthogonalization matrix S. This orthogonalization matrix does not introduce any computational load because it is merged during the quantization step.

Proposed Method

The proposed architecture encompasses a two parallel DCT blocks which are pipelined using a transposition buffer. The Fig. 3 shown below represents the proposed architecture. The two 1D DCT blocks are modeled using 12 additions which are discussed in section 3.1 and the transposition buffer in section 3.2

1D DCT Block:

Each 1D DCT block is constructed by appropriately placing 0's and 1's in the signed DCT transform which was introduced by Haweel [8]. The signed DCT uses signum function operator where T_{SDCT} is given as

$$T_{SDCT} = \frac{1}{\sqrt{N}} sign\{T_{SDCT}(u,v)\} \tag{4}$$

Where sign $T_{DCT}(u,v) = \{.\}$, which is given by

$$Sign\{a\} = \begin{cases} +1 & if\ a > 0 \\ 0 & if\ a = 0 \\ -1 & if\ a < 0 \end{cases} \tag{5}$$

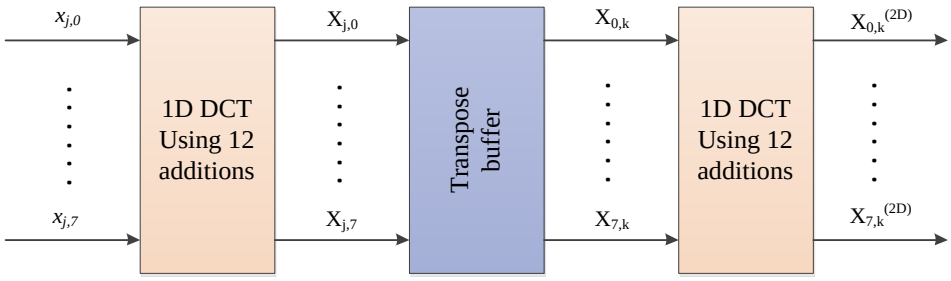

Figure 3: Block diagram of 2D DCT

The signed DCT transform comprises of only 0 or ±1 which removes the necessity of multiplication operation there by reducing the complexity of the architecture. It also provides good energy compaction and de-correlation characteristics. Several new approaches have been done to improve the PSNR values where the SDCT matrix is taken into consideration and the complexity has been reduced from 16[6] to 14[7] and finally to 12 additions by adding 0's and 1's to the SDCT matrix. Without providing any disturbance to the PSNR values the use of additions has been reduced considerably. The 12 additions matrix is represented below

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The above matrix T can be represented using the following signal flow diagram

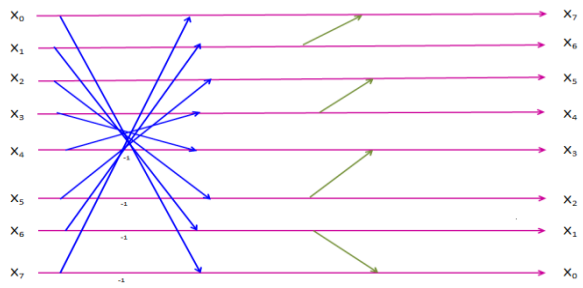

Figure 4: Signal flow diagram for DCT using 12 additions

From the above matrix T it can be concluded that the DCT transform can be obtained using 12 additions alone without any need of multipliers and bit shift operators as shown in Fig. 4. The diagonal matrix for the above matrix T is given as

$$D = \text{diag}(1, 1, 1, 1, 1, 1, 1, 1) * {}^{1}/_{2} \quad (6)$$

which can be merged with the quantization matrix which is already discussed in [4-7] to reduce the complexity during the computation of the DCT matrix.

Transposition Buffer:

The row wise 8x8 block matrix is computed using the first 1D DCT block which is transposed by the transpose buffer in Fig. 5, so that the second 1D DCT module can furnish column wise computation. In the transposition buffer each multiplexer is designed in such a way that it selects the respective row elements assigned to it there by creating the column. The 64 coefficients of the 8x8 matrix is fed into the transpose buffer of which the first element of each row is selected using the multiplexers and is fed as input to the next DCT module. In this way all the elements of each row are transposed into columns.

The proposed DCT architecture is pipelined using the transpose buffer in order to achieve higher speed with a small trade-off in area.

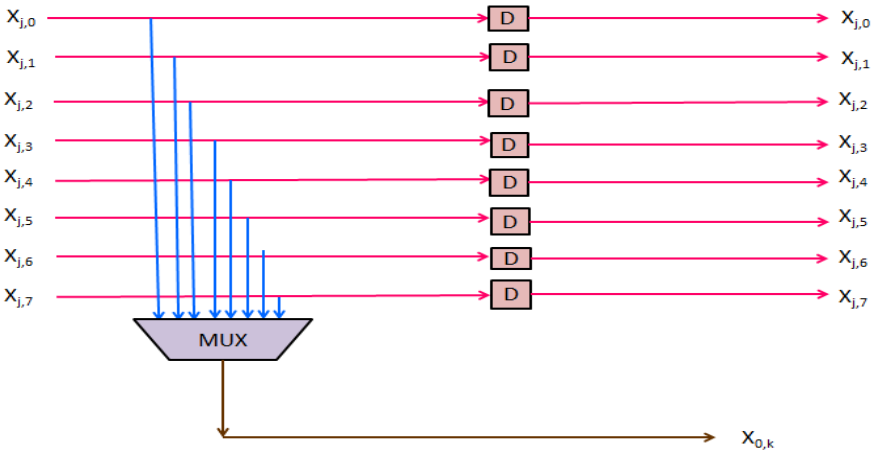

Figure 5: Transposition Buffer

Results and Discussions

Table 1: Timing Report

TIMING REPORT				
	Standard DCT	Bayer and Cintra[4]	Vaithiyanathan[6]	Proposed method
TIMING	4.361 ns	3.492 ns	3.429 ns	3.367 ns

Table 2: Area Report

AREA REPORT				
	Standard DCT	Bayer and Cintra[4]	Vaithiyanathan[6]	Proposed method
No: of slices	141	145	145	143
No: of slice flip flops	188	190	187	184
No: of 4-input LUTs	154	120	115	100
No: of bonded IOBs	24	144	144	144
No: of GCLKs	3	1	1	1

Conclusion

In this paper various 1D-DCT architectures are compared based on their performance and a novel method for reducing the number of additions from 14 to 12 in the 1D DCT module has been discussed. The 2D DCT architecture is pipelined to achieve a high throughput while employing a transpose buffer. The performance of the proposed 1D DCT architecture is calculated using Xilinx tool and the synthesis report was compared with the existing methods. Based on table 1 and 2 it can be concluded that the proposed 1D DCT architecture outperforms all the existing architectures in terms of both complexity and speed.

References

[1] Ahmed, N., Natarajan, T., & Rao, K. R. “Discrete cosine transform”. IEEE transactions on Computers, (1974)

[2] W. Chen, C. Smith, and S.C. Fralick, “A fast computational algorithm for the discrete cosine transform” IEEE Trans. On communication, (1977)

[3] RJ Cintra, FM Bayer, A DCT approximation for image compression. IEEE Signal Proc Let 18(10), (2011).

[4] FM Bayer, RJ Cintra, "DCT-like transform for image compression requires 14 additions only". Electron Lett 48(15), (2012).
[5] D Vaithiyanathan, R Seshasayanan, Low power DCT architecture for image compression, in Proceeding of the International Conference on Advanced Computing and Communication Systems (2013)
[6] D Vaithiyanathan, R Seshasayanan, S Anith, K Kunaraj, A low-complexity DCT approximation for image compression with 14 additions only, in Proceeding of the International Conference on Green Computing, Communication and Conservation of Energy (2013)
[7] TI Haweel, A new square wave transform based on the DCT. Signal Process (2001).
[8] Vijay Kumar Sharma, "Efficient VLSI Architectures for Image Compression Algorithms", NIT thesis, (2012).
[9] Uma Sadhvi Potluri, Arjuna Madanayake, , Renato J. Cintra, Fábio M. Bayer, Sunera Kulasekera and Amila Edirisuriya "Improved 8-Point Approximate DCT for Image and Video Compression Requiring Only 14 Additions" IEEE Transactions On Circuits And Systems,(2014)
[10] KA Wahid, M Martuza, M Das, C McCrosky, Efficient hardware implementation of 8x8 integer cosine transforms for multiple video codecs. J Real-Time Image Proc, (2013).
[11] PK Meher, SY Park, BK Mohanty, KS Lim, C Yeo, Efficient integer DCT architecture for HEVC. IEEE Trans on Circuits and Systems for Video Technology (2014).